Parametric Fuzzy Modelling Framework for Complex Data–Inherent Structures

Arne–Jens Hempel Steffen F. Bocklisch
Department of Electrical Engineering, Chair for Systems Theory, Chemnitz University of Technology
Chemnitz, Germany
Email: arne-jens.hempel@etit.tu-chemnitz.de, steffen.bocklisch@etit.tu-chemnitz.de

Abstract— *The present article dedicates itself to fuzzy modelling of data–inherent structures. In particular two main points are dealt with: the introduction of a fuzzy modelling framework and the elaboration of an automated, data–driven design strategy to model complex data–inherent structures within this framework.*

The innovation concerning the modelling framework lies in the fact that it is consistently built around a single, generic type of parametrical and convex membership function. In the first part of the article this essential building block will be defined and its assets and shortcomings will be discussed.

The novelty regarding the automated, data–driven design strategy consist in the conservation of the modelling framework when modelling complex (nonconvex) data–inherent structures. Instead of applying current clustering methods the design strategy uses the inverse of the data structure in order to created a fuzzy model solely based on convex membership functions.

Throughout the article the whole model design process is illustrated, section by section, with the help of an academic example.

Keywords— pattern recognition, fuzzy classification, fuzzy modelling of data structures, data–driven fuzzy classifier design

1 Introduction

Nowadays the world suffocates in vast amounts of data. To give those data an interpretable and thus economical meaning it has to be analysed. The goal of such an analysis is the creation of a model or the classification of the considered phenomenon, e. g. modelling of the traffic flow in cities, medical or machine diagnosis [1, 2, 3].

Basically there are two main philosophies to deduce such a model, theoretical and experimental modelling. In experimental modelling it is assumed that measurement data (objects) reflect the complexity of the phenomenon under consideration through data–inherent structures. Unfortunately, the same data might also exhibit imprecision (e. g. measuring inaccuracies) or depict interesting phenomena characteristics just vaguely (because of missing information). With the help of fuzzy set theory these occurring inaccuracies can be taken into account as a supplementary model feature [4].

In this work the whole modelling problem is understood as a fuzzy classification task, where specific fuzzy sets form a model equivalent. As it is pointed out in [5], there are a lot of sophisticated solutions for such a task, which in general apply nonparametric fuzzy sets or a composition of different fuzzy sets. Contrary to those approaches, the main philosophy behind this work is the exclusive usage of one specific parametrical fuzzy set to model complex data–inherent structures as well as the data itself. Another aspect of the here pursued type of structure modelling is that it works in the original feature space without any transformation like fuzzy support vector classifiers or any assumption of fuzzy functions [6, 7].

2 Fuzzy Pattern Classes

In order to become acquainted with the modelling philosophy it is necessary to understand its core component, the so called *fuzzy pattern class* (FPC). The subsequent sections provide a basic survey about the definition, composition, capabilities and utilisation of *fuzzy pattern class models*.

From a fuzzy theoretical perspective fuzzy pattern classes correspond to a side–specific parametrical multivariate membership function. FPCs are referred to as classes since they emerge from an agglomeration of class supporting objects (see section 2.2), consequently they represent an superordinate entity.

2.1 Definition of a Fuzzy Pattern Class

Although the usual FPC–membership function is multidimensional it derives from one–dimensional basis functions. Hence, it is reasonable to study this basis functions being equivalent to one–dimensional fuzzy pattern classes first. Generally a one–dimensional fuzzy pattern class A is defined over its class space U based on a side–specific parametrical function concept, see (1).

$$\mu^A(a,\vec{p}) = \begin{cases} \dfrac{a}{1+\left(\dfrac{1}{b_l}-1\right)\left|\dfrac{u}{c_l}\right|^{d_l}}, & u<0 \\[2ex] \dfrac{a}{1+\left(\dfrac{1}{b_r}-1\right)\left|\dfrac{u}{c_r}\right|^{d_r}}, & u\geq 0 \end{cases} \tag{1}$$

The function concept comprises a set of seven parameters a and $\vec{p} = (b_l, b_r, c_l, c_r, d_l, d_r)$. The further specification of these parameters results from the fact that the parameter a characterises an entire fuzzy pattern class, whereas the parameters combined in $\vec{p}$ are related to a dimension of the class space [8]. Beyond their mere mathematical functionality all parameters possess the following semantical meaning:

- The parameter a represents the maximum membership value of the FPC μ^A. Regarding a structure of classes the parameter a expresses the weight of a specific class. Considering a dynamic classification process a embodies the topicality or authenticity of the information represented by that class [9, 10].
- In the normalised case $a = 1$, the parameters b_l, b_r of $\vec{p}$ assign left– and right–sided membership values at the class borders $u = -c_l$ and $u = c_r$.
- c_l, c_r mark the support of a class in a crisp sense. Both parameters characterise the left– and right–sided expansions of a fuzzy pattern class.
- The continuous descent of the membership function is specified by the parameters d_l, d_r. From a graphical point of view d_l, d_r determine the shape of the membership function, or in

other words, the fuzziness of a class. Fig. 1 illustrates the introduced concept of the membership function considering the general unidimensional case.

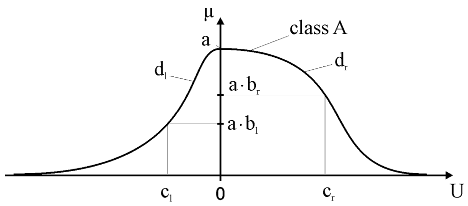

Figure 1: Membership function and parameters.

To obtain the common multidimensional fuzzy pattern class A the basis functions of each class dimension are accumulated using the N-fold compensatory Hamacher intersection operator (2), where n denotes the index of the basis functions and N the total number of dimensions [11].

$$k_{Ham\cap}\,\mu_N^A = \frac{1}{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{\mu_n^A}} \tag{2}$$

Regarding the main philosophy behind this paper, the most important feature of this intersection operation is the conservation of the parametrical function concept for the multidimensional case [11].

Considering the multidimensional FPC form the class describing set of parameters is supplemented by a class specific position $\vec{u}_0$ in the original feature space and a class specific orientation $\vec{\varphi}$. Fig. 2 depicts the influence of the additional parameters for a two dimensional three class structure. The different location of each class results from $\vec{u}_{c10} = (0.8, 0.2)^T$ $\vec{u}_{c20} = (0.5, 0.5)^T$ and $\vec{u}_{c30} = (0.2, 0.8)^T$, whereas an additional class orientation φ of 60° has been applied to the middle class.

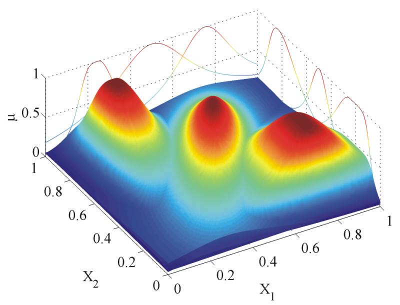

Figure 2: Two dimensional three class structure

2.2 *Data Driven Design of Fuzzy Pattern Classes*

Being familiar with the definition of fuzzy pattern classes a more intriguing question is how fuzzy pattern class models can be deduced. As a matter of principle such models can be obtained via two different approaches [8].

First they can be defined by expertise. That is an expert determines all class parameters based upon task and domain specific knowledge. This approach is not pursued here.

The second approach is a data–driven method, strongly advocating the here featured goal to model data–inherent structures. Based upon a class labelled set of learning data x_l (e.g. measurement objects) all class parameters are assigned automatically by a two step aggregation procedure [9, 10]. The class labels might result from a preliminary conducted cluster analysis.

For the sake of clarity only the basic aggregation principle will be outlined in the following. A full description can be found in [9, 12]. In order to perform the aggregation on sound mathematical foundations, the crisp learning dataset is extended to a set of fuzzy objects, using the introduced function concept (1). This extension is justified by the fact that every observation (measurement) inheres a so called "elementary fuzziness" (e.g impression of a sensor) [9]. In the first aggregation step the class position $\vec{u}_0$, alignment φ and extensions c_l, c_r are calculated in a dimension–wise manner. As exemplified in Fig. 3 the position $\vec{u}_0$ and alignment $\vec{\varphi}$ of the class space is obtained via a principal component analysis (PCA), where $\vec{u}_0$ is defined by the mean over all learning objects and $\vec{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_{N-1})^T$ by sequential rotation of the class space U into the principal axes.

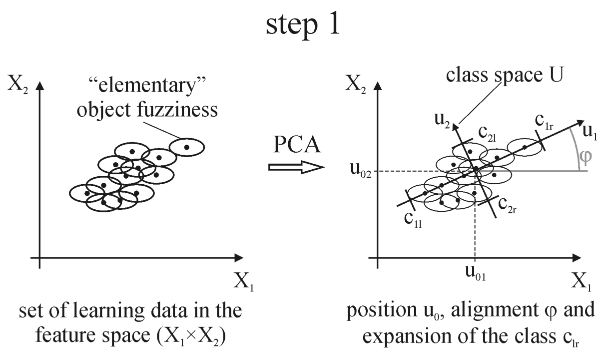

Figure 3: Step 1: aggregation procedure

The extensions c_l, c_r of the class are determined by the outermost objects in each class space dimension. In the subsequent second aggregation step the class shape d_l, d_r and the border memberships b_l, b_r are determined also dimension–wisely.

After the transformation of the objects x_l into their corresponding class space U the shape of a fuzzy patter class (d_l, d_r) is assigned based on their agglomeration properties. The more the data resembles an agglomeration according to a geometric series the smoother the class shape. The rate of resemblance is determined by the mean distance between two adjacent objects. The smoothest class shape is obtained for $d_{lr} = 2$ where the objects are cumulating in the centre of the class conform to a geometric series. The crisp case results for ($d_{lr} \to \infty$) where the objects are equally distributed over the class space, however for calculation purposes $d_{lr} = 20$ has proven to be a sufficient value to represent the crisp case.

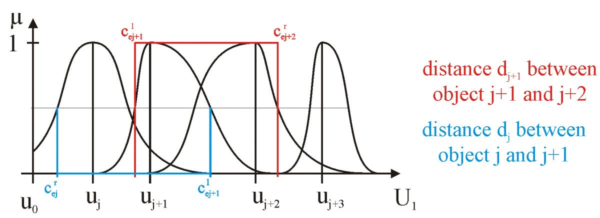

Figure 4: Step 2a: determination of the class form

The values for border memberships b_l, b_r are derived by the conservation of the object cardinality taking into account the results of c_{lr} and d_{lr}. Since the integral over the class membership function (1) cannot be solved analytically the border memberships b_l, b_r are estimated by a binary search over the unity interval $b_{lr} \in [0,1]$ [9].

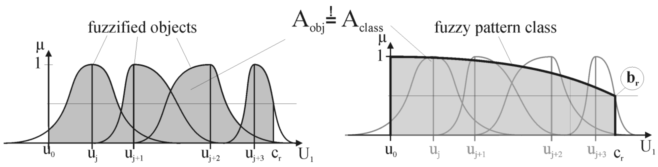

Figure 5: Step 2b: claim for cardinality

The class "weight" a is calculated by a logistic function given the number of the class supporting objects, see [9].

2.3 *Application of FPC and Mode of Operation*

For applicative purposes all task relevant fuzzy pattern classes are grouped together in a so called fuzzy pattern classifier. In operating mode the fuzzy pattern classifier assigns unknown objects to this class structure. The objects to be classified are each denoted by a vector $\vec{x}$ of their features:

$$\vec{x} = (x_1, x_2, \ldots, x_N)^T, \tag{3}$$

where N represents the number of feature dimensions. The results of the classification process are stored into a so called vector of sympathy $\vec{s}$. The components of $\vec{s}$ express the membership of a classified object to the corresponding class:

$$\vec{s} = (s_1, s_2, \ldots, s_K)^T, \tag{4}$$

where K is the total number of classes. The gradual membership of an object to a given class is calculated using (1).

$$s_k = \mu^k(\vec{x}) \quad \text{for } k = 1, 2, \ldots, K \tag{5}$$

Figure 6 illustrates the process of classification with the help of a one-dimensional three class structure. The object to be classified is situated in the centre of class two, the right outskirts of the first class and in the left centre of the third class. Alongside with the classification task the classification results are listed.

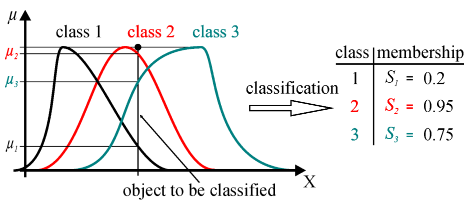

Figure 6: Object classification

According to Fig.6 the vector of sympathy describes a unique assignment of the object to the class structure with respect to its location in the feature space, since there are three classes it contains three values of membership.

2.4 *Properties of Fuzzy Pattern Classes*

In order to round off the comprehension about the afore introduced concept of fuzzy pattern class models its major properties (advantages and drawbacks) will be outlined subsequently. The main features of the fuzzy pattern class model lie in its versatility, its uniformity and its closed modelling framework. All these features can be attributed to the unimodal, side–specific and parametric class membership function. In the most general case the class membership function offers multivariate FPC models with various and asymmetric shapes, ranging from peak- over bell- to crisp shaped fuzzy sets. In connection with the introduced data–driven design procedure the class membership function allows to map class internal object distributions onto its shape and to model correlative relations without loosing its fuzzy logic basis. Besides this flexibility it has to be stressed that the parameters are semantically motivated or have at least a semantical meaning. It is therefore that the fuzzy pattern classes are considered to be well interpretable and transparent. Another feature resulting from the utilisation of the parametric membership concept is its good trade off between data compression, computational cost and generality. Due to the choice of the membership function (1) each fuzzy pattern class is defined on a set of eight parameters per dimension, providing a sufficient data compression, especially for high dimensional models. Due to the choice of the conjunction operator (2) the intersection operation is exclusively performed on *parameter level* saving computational cost. Both advantages are traded off for generality in so far as class membership functions are convex models, specifying a convex area of the feature space. Consequently FPC are best suited to model convex data–inherent structures. But their convex nature causes fuzzy pattern class models to be afflicted with significant errors when it comes to model nonconvex data–inherent structures.

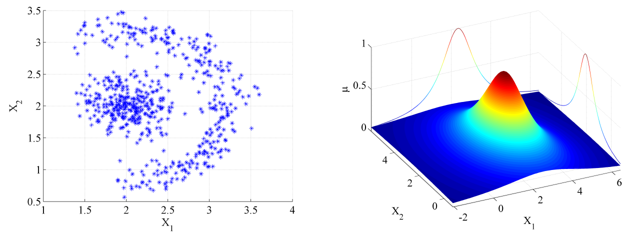

Figure 7: Nonconvex data structure and according FPC model

An example for such an error is depicted in Fig.7 where a central object accumulation enclosed by a half circle shaped data structure was aggregated to a fuzzy pattern class. Obviously the region between the object accumulations does not belong to the given data structure but the associated fuzzy pattern model μ^C will assign high grades of memberships for this region.

In order to circumvent this major drawback two possibilities can be thought of:
The first way, is to segment the data into convex subsets, for example with the help of cluster algorithms. Aside from the fact that this approach works on every data–inherent structure it might create considerably large structures of fuzzy pattern classes at the expenses of model clarity and computational costs.

3 Fuzzy Pattern Anti–classes

The hereafter elaborated access to dissolve the convexity drawback arises from the negation of a class assertion over its unsupported class space. The idea behind this approach

can be stated as follows: *The difference of convex sets can be a non–convex set.*
Fig.8 sketches such a negation of the FPC model for a ring shaped problem.

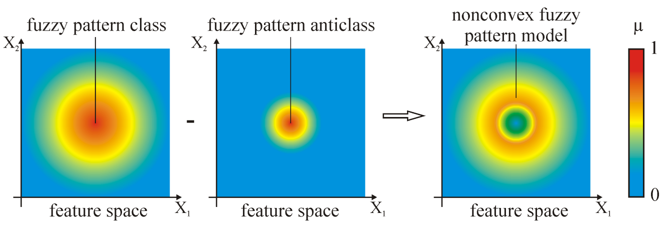

Figure 8: Nonconvex fuzzy pattern model via negation

As it is depicted the negation will be introduced in terms of fuzzy pattern anti–classes (FPAC). It works on semantical level and from this point of view FPACs can be seen as a further specification of a preceding FPC.
Mathematically fuzzy pattern anti–classes are defined upon the same membership function concept as fuzzy pattern classes, see (1). The main concern of of this definition is conservation of the fuzzy pattern modelling framework and with it the *automated model generation* and the *model properties* (such as flexibility, interpretability, computational efficiency, etc.). The enormous shape diversity of fuzzy pattern classes together with the mutual negation of such membership functions allows to model almost any form of data–inherent structures. The only task that needs to be solved can be formulated as follows:
Determine the fuzzy pattern anti–classes given a set of learning data, containing an arbitrary (nonconvex) data–inherent structure and the appendent FPC model.

3.1 Design of Fuzzy Pattern Anti–classes

Assuming that FPACs, like usual fuzzy pattern classes, can be supported by objects or, better so called "anti–objects", then it is in accordance with modelling framework that FPCAs can be designed in a data–driven manner. Aiming to elaborate an automated FPAC design method the already introduced automated databased algorithm can be exploited. However the introduced data–driven FPC design relies on class supporting objects. Correspondingly, in order to setup FPACs these "anti–objects" have to be found first. When considering the general case there are no "anti–objects" given. Consequently they have to be *generated*.
For the sake of clearness the anti–object generation procedure will be illustrated for the two–dimensional case with the help of an academic example. The general multidimensional case follows analogously.

3.1.1 Generation of Anti–objects

According to the task (section 3) the information given are the location of each class supporting object in the learning dataset and the modelling fuzzy pattern class, see Fig.7. Suppose that there is no prior information about the distribution of the learning data it is impossible to make an assumption about the shape of a data–inherent structure. That is why the central idea behind the anti–object generation is led by the assumption that the *anti–objects will agglomerate in the class space being unsupported by learning objects*, such that they will adopt a kind of inverse data–inherent structure and form FPACs.
To ascertain whether a partition of the class space is actually supported by learning objects requires a discretisation of the class space. The size of the class space to be discretised is determined based upon the class borders c_{lr} in each dimension, whereas the discretisation resolution amounts to 2% of the class space leading to a 50×50 matrix F with 2500 cells. A cell of an N–dimensional class space then posses the following the extend:

$$V_c = \frac{1}{50}\left((c_{l1}+c_{r1})\cdot(c_{l2}+c_{r2})\cdot\ldots\cdot(c_{lN}+c_{rN})\right) \qquad (6)$$

Fig.9 depicts the discretised class space for the example along with the class borders in green, the object supported cells highlighted in red and the unsupported class space cells coloured in dark blue.

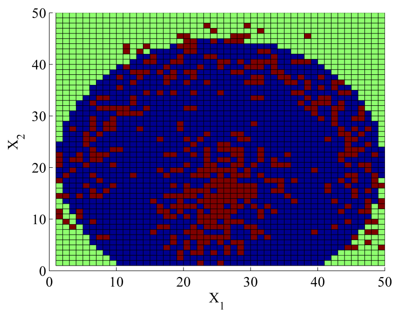

Figure 9: Discretised class space for the example

According to the central idea, the anti–objects have to agglomerate in these unsupported cells. In other words it is necessary to define an model for the agglomeration process. In particular two ways have been pioneered to model the accumulation of anti–objects, namely accumulation based on the elemental fuzziness and accumulation based on power series expansion. The most promising agglomeration model resulted from the assumption of a Fibonacci series.

Its leading thought is the expansion of Fibonacci numbers around the cell of interest until an object supported cell or a class border cell is reached. The result of the expansion around an arbitrary cell (i, j) is a 50×50 matrix $F_{d(i,j)}$ containing only the expanded Fibonacci numbers and zeros.

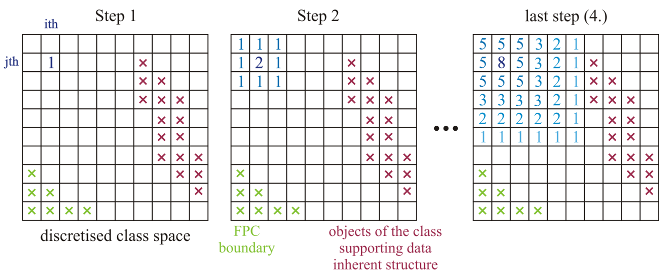

Figure 10: Fibonacci expansion for an arbitrary cell

In this context it has to be mentioned that the fuzzy pattern class borders are treated like objects. Purpose of this procedural manner is to provide an insight of the inner class anti–object distribution by preventing an anti–object accumulation close to the class borders.

Fig.10 illustrates the implementation of the Fibonacci expansion for an arbitrary cell at position (i,j). The sum over all Fibonacci matrices $F_{d(i,j)}$ yields the result of the agglomeration process, as in (7).

$$F_d = \sum_{i=1}^{N} \sum_{j=1}^{N} F_{d(i,j)} \tag{7}$$

F_d can be interpreted as a kind of anti–object density matrix. It is broken down into an explicit number of anti–objects per cell based on the maximum object density of all cells d_{max}, see (8).

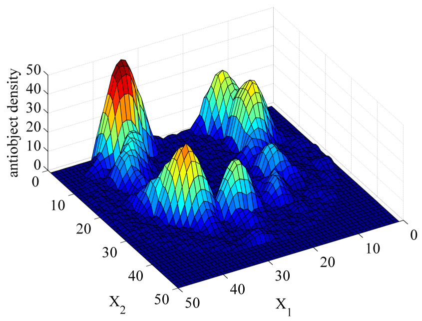

Figure 11: Anti–object density matrix F_d for the example

$$F_a = rd\left(\left(\frac{d_{max}}{max\left(F_d\right)}\right)F_d\right) \tag{8}$$

With the usual rounding operation rd, in (8), it is assured that each cell contains a natural number of anti–objects ($F_a \in \mathbb{N}^{50\times 50}$).

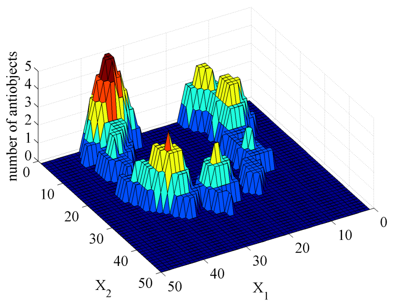

Figure 12: Anti–object matrix F_a for the example

Fig.12 presents the resulting number of anti–objects in their corresponding cells, after applying (8), for the example. The total number of anti–objects, being distributed around the centre region amounts to 887 with a maximum density of five anti–objects per cell.

The final step to complete the anti–object generation comprises the assignment of a position to each anti–object. For this purpose all anti–objects are uniformly distributed over their associated cell. In the here discussed two–dimensional case the anti–object position of an arbitrary cell with the coordinates (i,j) follows from (9), where *rand* draws an random number from the unity interval.

$$\begin{pmatrix} u_{a1} \\ u_{a2} \end{pmatrix} = \begin{pmatrix} i \cdot \frac{1}{50}\left(c_{l1}+c_{r1}\right) - rand \\ j \cdot \frac{1}{50}\left(c_{l2}+c_{r2}\right) - rand \end{pmatrix} \tag{9}$$

After their distribution the set of anti–objects forms an inverse of the data–inherent structure, see left–hand side of Fig.13.

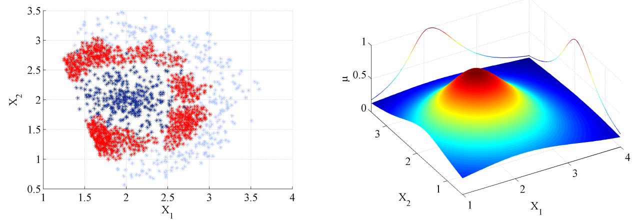

Figure 13: left: Objects, anti–objects, right: according FPAC

However, as it can be seen the anti–objects (in red) form a nonconvex data structure. Modelling this with the help of the automated FPC algorithm again creates a faulty FPAC μ^{AC}, see right–hand side of Fig.13. This might lead to the misconception that the problem was just shifted to the anti–objects. But that is by no means the case, since the whole procedure can be applied again to the created anti–class.

Like a block letter and its print anti–objects and objects complement each other to a complete convex model. Or with another emphasis anti–objects together with objects cover the *entire* class space. Because of that complementary relation, the original objects can be seen as the anti–objects of the anti–objects. Consequently, instead of running through the entire process again for the anti–class, the original objects can be recycled. The reutilisation of the original objects is justified by the facts that a part of the original objects is already complementing the set of anti–objects to a convex description and that their position is already known. Original objects being considered to negate the anti–class are required, firstly to posses a high membership to the anti–class and secondly to be located within the class borders of the anti–class. Generally this applies to original objects with a anti–class membership above $\mu = 0.5$.

For the considered example, the only objects meeting this requirement are the ones located in the centre region, highlighted in dark blue see Fig.13. After setting up the "anti–anti–class" μ^{AAC} by the selected original objects the negation procedure stops owing to the fact the average membership of possible anti–objects drops below a threshold of $\mu = 0.5$.

3.2 Combination of Fuzzy Pattern Anti–class Models

After their generation the anti–class models have to be combined together in order to from an overall model of the data–inherent structure. This combination process is characterised by the following keynote: *If an object does not belong to the anti–class it belongs to the preceding class.* A reasonable implementation of this key concept derives from the concatenation of the natural complement and minimum conjunction [13, 14].

$$\mu = min\left(\mu^{class}, \left(1 - \mu^{anti-class}\right)\right) \tag{10}$$

If the anti–class generation yields several levels of negation their sequence has to be respected for the concatenation. Or differently it has to be respected that *each level of negation creates an new level of hierarchy*. Equation (11) represents this aspect for the example.

$$\mu = min\left(\mu^{C}, 1 - min\left(\mu^{AC}, \left(1 - \mu^{AAC}\right)\right)\right) \tag{11}$$

Fig.14 depicts the nonconvex and fuzzy model, resulting from the class–anti–class combination, together with the supporting objects in black.

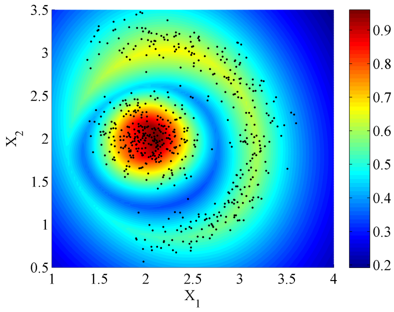

Figure 14: Resulting FPC Model

3.3 Properties of the Fuzzy Pattern Anti–class Design

The introduction of FPAC design is round off by having its assets and drawbacks summarised subsequently.

The main advantage of the introduced design approach lies in the exploitation of a single type of membership function. All the beneficial properties of fuzzy pattern classes, such as transparency, interpretability, computational efficiency etc. are imparted and conserved into the design of nonconvex FPC models. Equally important, by complementing the objects over the considered class space the FPAC design converts the major drawback of fuzzy pattern models (convexity) into an advantage. As a consequence the anti–objects have to be constructed only once. Furthermore the complete object anti–object covering of the class space guarantees convergence of the FPAC design method, since there is no further anti–object generation. Besides that, each anti–class is smaller or equal in size compared to its preceding class such that the area of interest is also converging to zero. Another aspect worth to mention is that the FPAC design works independent from clustering algorithms but features structure capturing.

The major drawback of the design strategy arises from its numerical character, the class space has to be discretised in order to distribute the anti–objects. As a consequence their creation is computationally costly.

4 Conclusions

This paper presents a data driven approach toward modelling of complex data–inherent structures. Its main philosophy is the exclusive usage of a general fuzzy modelling framework.

Instead of applying cluster analysis techniques the design strategy aims to complete the convex fuzzy pattern model with the help of so called anti–objects. These anti–objects are not available prior to the design. They have been generated and distributed over the class space. For this purpose a Fibonacci expansion model was elaborated and demonstrated.

With the help of the introduced automated FPC design the anti–objects have been agglomerated to a negating fuzzy pattern classes (FPAC). By preserving the membership function concept the FPAC are on one hand afflicted again with the drawbacks of fuzzy pattern classes. But on the other hand the same design approach can be applied again, eventually leading to a pure convex fuzzy pattern model. A possible combination of the setup FPACs and the original fuzzy pattern model to a hierarchical nonconvex overall FPC model has been demonstrated with the help of an example.

Finally it has to be stressed that the introduced design strategy is a universal approach in so far as it can be applied to any multivariate unimodal convex parametrical membership function.

Further points of research are:

- reducing the numerical character of the method
- combination with cluster algorithms
- creation of fuzzy pattern class and anti–class networks

References

[1] Gernot Herbst and Steffen F. Bocklisch. Classification of keystroke dynamics - A case study of fuzzified discrete event handling. In Bengt Lennartson, Martin Fabian, Knut Akesson, Alessandro Giua, and Ratnesh Kumar, editors, *9th International Workshop on Discrete Event Systems*, 2008.

[2] Arne-Jens Hempel and Steffen F. Bocklisc. Hierarchical modelling of data inherent structures using networks of fuzzy classifiers. In *Tenth International Conference on Computer Modeling and Simulation, 2008. UKSIM 2008.*, pages 230–235. IEEE, 2008. ISBN: 0-7695-3114-8.

[3] C. Weihs & W. Gaul, editor. *Classification - the Ubiquitous Challenge: Proceedings of the 28th Annual Conference of the Gesellschaft Fur Klassifikation E.V., University of Dortmund*, volume 28. Springer, June 2005.

[4] Lotfi Asker Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[5] James C. Bezdek, James Keller, Raghu Krisnapuram, and Nikhil R. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing (The Handbooks of Fuzzy Sets)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[6] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, 2006.

[7] Xuehua Li and Lan Shu. Fuzzy theory based support vector machine classifier. *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, 1:600–604, 2008.

[8] Steffen F. Bocklisch. *Prozeßanalyse mit unscharfen Verfahren*. VEB Verlag Technik, Berlin, 1987.

[9] Arne-Jens Hempel. Aggregation und Identifikation von Fuzzy−Objekten mit unterschiedlichen elementaren Unschärfen. Master's thesis, TU Chemnitz, 2005.

[10] Michael Päßler. Mehrdimensionale Zeitreihenmodellierung und Prognose mittels Fuzzy Pattern Modellen. Master's thesis, TU Chemnitz, 1998.

[11] Ulrich Scheunert. *Neue Verfahren der Fuzzy−Mengen−Verknüpfung und Fuzz−Arithmetik und ihrer Anwendung bei der Sensor−Daten−Fusion*. PhD thesis, TU Chemnitz, 2001.

[12] Steffen F. Bocklisch and Norman Bitterlich. *Fuzzy systems in computer science*, chapter Fuzzy pattern classification – methodology and application. Vieweg, Wiesbaden, 1994.

[13] Leszek Rutkowski. *Flexible Neuro-fuzzy Systems: Structures, Learning and Performance Evaluation (Kluwer International Series in Engineering and Computer Science)*. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[14] H. J. Zimmermann. *Fuzzy Set Theory and its Applications*. Springer, 4 edition, October 2001.